

化工应用数学 第二章 python 编程基础 讲义

(01.Python 基础语法)

TIOBE 编程语言排行及 Python 表现:

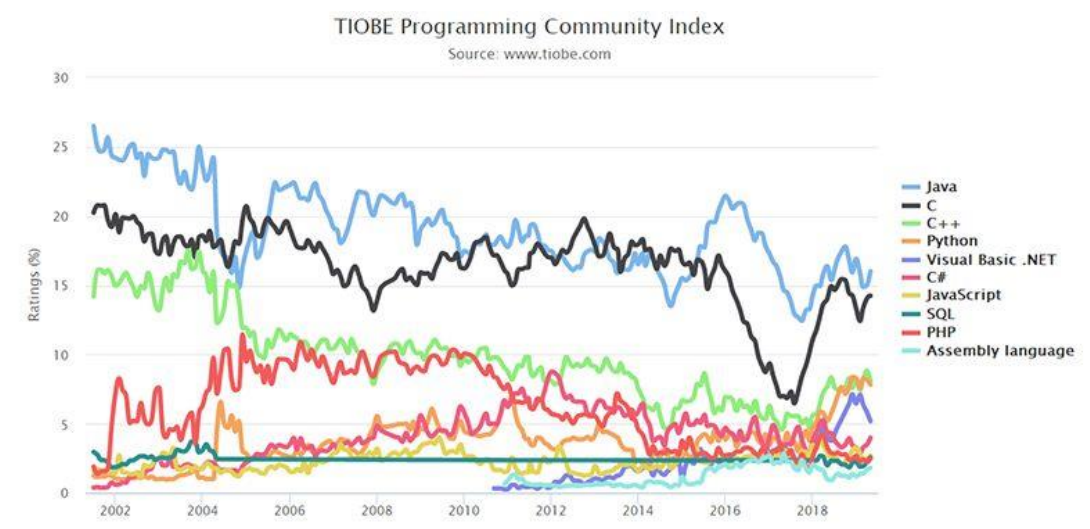
下图为 TIOBE 编程语言社区 2019 年 5 月排行榜的前 20 名情况，可以看出 Python 依然是编程语言界的“快跑者”，增幅从上个月的 2.36% 上升到 2.64%

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	14.243%	+0.24%
3	3		C++	8.095%	+0.43%
4	4		Python	7.830%	+2.64%
5	6	▲	Visual Basic .NET	5.193%	+1.07%
6	5	▼	C#	3.984%	-0.42%
7	8	▲	JavaScript	2.690%	-0.23%
8	9	▲	SQL	2.555%	+0.57%
9	7	▼	PHP	2.489%	-0.83%
10	13	▲	Assembly language	1.816%	+0.82%
11	15	▲	Objective-C	1.626%	+0.69%
12	12		Delphi/Object Pascal	1.406%	+0.39%
13	18	▲	Perl	1.394%	+0.48%
14	16	▲	MATLAB	1.366%	+0.44%
15	10	▼	Ruby	1.343%	+0.16%
16	17	▲	Visual Basic	1.317%	+0.40%
17	91	▲	Groovy	1.173%	+1.06%
18	19	▲	Swift	1.150%	+0.24%
19	14	▼	Go	1.114%	+0.14%
20	22	▲	PL/SQL	1.017%	+0.12%

其中，其中主要语言的历史排名情况如下表

Programming Language	2019	2014	2009	2004	1999	1994	1989
Java	1	2	1	1	13	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	2
Python	4	7	5	9	26	21	-
Visual Basic .NET	5	10	-	-	-	-	-
C#	6	5	6	7	23	-	-
JavaScript	7	8	8	8	18	-	-
PHP	8	6	4	5	-	-	-
SQL	9	-	-	6	-	-	-
Objective-C	10	3	34	42	-	-	-
COBOL	25	20	16	11	3	9	12
Lisp	30	13	19	14	11	5	3
Pascal	212	15	14	96	6	3	19

而针对 Top 10 编程语言，其历史发展趋势如下图所示：



Python 来源：

Python 由荷兰人 Guido van Rossum(GvR)于 1989 年发明，而他之所以取“Python”的名字是因为 GvR 是一个蒙提·派森（Monty Python）的飞行马戏团的狂热爱好者。

GvR 设计 Python 的动机：

在 80 年代，个人电脑的配置较之如今是极低的，程序员不得不努力思考如何最大化利用空间，以写出符合机器口味的程序。而正是因为这一点，让 Guido 感到苦恼。他认为这样编写程序实在是太过于耗费时间，于是他想到了 shell。shell 可以像胶水一样，将 UNIX 下的许多功能连接在一起。许多 C 语言下上百行的程序，在 shell 下只用几行就可以完成。然而，shell 的本质是调用命令，它并不是一个真正的语言，shell 不能全面的调动计算机的功能。

于是，Guido 开始思考，是否能设计一款语言，使它同时具备 C 与 shell 的优点，既能够全面调用计算机的功能接口，又可以轻松编写程序。

后来他进入 CWI（Centrum Wiskunde & Informatica，数学和计算机研究所）工作，并参加了 ABC 语言的开发。ABC 语言旨在让语言变得容易阅读，容易使用，容易记忆，容易学习。但由于它的可拓展性差，不能直接 IO，过度革新与传播困难，导致它不为大多数程序员所接受（因为语法上的过度革新，加大了程序员的学习难度）与传播。

除去这些缺点，我们不难看出，ABC 就 Guido 心中所期望的那款语言的雏形。1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。而取名 python，是取自他挚爱的一部电视剧 Monty Python’s Flying Circus。

1991 年，第一个 Python 编译器诞生，它是用 C 语言实现的，并能够调用 C 库(.so 文件)。从一出生，Python 已经具有了：类(class)，函数(function)，异常处

理(exception)，包括表(list)和词典(dictionary)在内的核心数据类型，以及模块(module)为基础的拓展系统。

Guido 为防止重蹈 ABC 的覆辙，着重注意 python 的可扩展性，并且也沿用了 C 中的大部分语法习惯。

Python 应用举例：

Google: Python 用于很多谷歌项目网络应用的后端，如 Google Groups、Gmail、Google Maps 等，Google APP Engine 支持使用 Python 作为开发语言

NASA: 美国宇航局从 1994 年起把 Python 作为主要开发语言

豆瓣网

Torchlight: Python 编写的大型 3D 游戏

Blender: 使用 C 和 Python 开发的开源 3D 绘图软件

Python 特性：

交互式命令：可以单步直译运行，运行 Python 解释器或者 IDLE 进入交互式命令行环境，在提示符号>>>后输入代码，按 Enter 执行

```
>>> print("Hello, Python!")
Hello, Python!
```

不只是脚本语言：脚本语言仅能支持简单编程任务，如 Lua、JavaScript 等，而 Python 是面向对象的，并且支持异常处理和类型检查

强大易用的标准库：如 Html/XML 解析 (BeautifulSoup、Expat)、字符串处理 (字典、数组)、正则表达式 (re)、单元测试 (PyUnit)、网络访问 (urllib2)、图形模块 (Tkinter、PyTCL、WxPython)

胶水语言：Python 可以用来将不同语言编写的程序“粘”在一起---胶水语言！比如可以使用 c 语言对于性能要求较高的部分，而使用 Python 调用相应的模块；比如 Boost.Python 用于 Python 和 C/C++ 的类库相互调用，Jython 用于 Java、而 IronPython 用于 .NET 语言。

```
4 #read data file
5 def readMatrix(fileName,separator,readFirstLine):
6     f = open(fileName)
7     lines = f.readlines()
8     if readFirstLine:
9         row = 0
10    else:
11        row = -1
12    RowNum = len(lines)+row
13    ColumnNum = len(lines[0].strip('\n').split(separator))
14    A = np.zeros((RowNum,ColumnNum),dtype=float)
15    for line in lines:
16        list = line.strip('\n').split(separator)
17        if readFirstLine:
18            A[row,:] = list[:]
19        else:
20            if row != -1:
21                A[row,:] = list[:]
22            row+=1
23    f.close()
24    return A,RowNum,ColumnNum
```

不需要括号：Python 使用缩进而不是括号来进行代码段的标识，减少了视觉上的混乱，并有效的缩短了程序的行数，从而提高了程序的可读性。

Python 环境搭建：

Python 的最新代码及相关资讯可以从其官网下载和浏览：

Python 官网：<https://www.python.org/>

相关文档说明可以从下方网址下载：

Python 文档下载地址：<https://www.python.org/doc/>

以 Window 平台上安装 Python 为例：

- 1.打开 WEB 浏览器访问 <https://www.python.org/downloads/windows/>
- 2.在下载列表中选择 Window 平台安装包,包格式为:python-XYZ.msi 文件, XYZ 为你要安装的版本号。
- 3.使用安装程序 python-XYZ.msi, Windows 系统必须支持 Microsoft Installer 2.0 搭配使用。只要保存安装文件到本地计算机,然后运行它,看看你的机器支持 MSI。Windows XP 和更高版本已经有 MSI,很多老机器也可以安装 MSI。
- 4.下载后,双击下载包,进入 Python 安装向导,安装非常简单,你只需要使用默认的设置一直点击"下一步"直到安装完成即可。



特别要注意选上 pip 和 Add python.exe to Path，否则后续需要重新设置环境变量。

安装完成之后，可通过开始菜单选择 Python 文件夹，进而选择相应程序快捷链接。其中：

IDLE: 图形界面

Python 3.6: 命令行形式

Python 3.6 Module Docs: 说明文本

Python 执行方式:

交互式: 不需要创建脚本文件, 是通过 Python 解释器的交互模式进来编写代码并运行

脚本式: 首先编写并创建脚本文件, 然后通过运行脚本文件来运行代码

Python 标识符:

在 Python 里, 标识符由字母、数字、下划线组成。

在 Python 中, 所有标识符可以包括英文、数字以及下划线(_), 但不能以数字开头。

Python 中的标识符是区分大小写的。

以下划线开头的标识符是有特殊意义的。以单下划线开头_foo 的代表不能直接访问的类属性, 需通过类提供的接口进行访问, 不能用 from xxx import *而导入。

以双下划线开头的__foo 代表类的私有成员, 以双下划线开头和结尾的__foo__代表 Python 里特殊方法专用的标识, 如__init__()代表类的构造函数。

Python 可以同一行显示多条语句, 方法是用分号分开, 如:

```
>>> print('hello');print('runoob');  
hello  
runoob
```

Python 保留字符:

保留字符是 Python 本身已经定义或者用到的一些字符, 它们本身在 Python 中具有特殊的含义。这些保留字不能用作常数或变数, 或任何其他标识符名称。

所有 Python 的关键字只包含小写字母。

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Python 中的其他基础符号:

行与缩进: Python 与其他语言最大的区别就是, Python 的代码块不使用大括号 {} 来控制类, 函数以及其他逻辑判断。python 最具特色的就是用缩进来写模块。缩进的空白数量是可变的, 但是所有代码块语句必须包含相同的缩进空白数量, 这个必须严格执行。

多行语句: 可以使用斜杠 (\) 将一行的语句分为多行显示

引号: 可以使用引号(')、双引号(")、三引号('' 或 ''') 来表示字符串, 引号的开始与结束必须是相同类型; 其中三引号可以由多行组成, 编写多行文本的快捷语法, 常用于文档字符串, 在文件的特定地点, 被当做注释。

注释: python 中单行注释采用 # 开头。在文档中, 也会采用上述三引号的方式来作为注释。

Python 变量类型:

创建变量时系统会在内存中开辟一个空间。基于变量的数据类型, 解释器会分配指定内存, 并决定什么数据可以被存储在内存中。变量可以指定不同的数据类型, 可以存储整数, 小数或字符。

Python 中的变量赋值不需要类型声明。每个变量在使用前都必须赋值, 变量赋值以后该变量才会被创建。Python 中使用等号(=)来给变量赋值, 等号(=)运算符左边是变量名, 等号(=)运算符右边是存储在变量中的值。

在 Python 里, 可以同时为多个变量赋值, 比如:

```
a = b = c = 1
```

甚至可以为多个对象指定多个变量, 如:

```
a, b, c = 1, 2, "john"
```

Python 中的五种标准数据类型: Numbers (数字)、String (字符串)、List (列表)、Tuple (元组) 和 Dictionary (字典)

数字类型用于存储数值, 其包括以下四种类型: int (有符号整型)、long (长整型)、float (浮点型) 和 complex (复数), 需要注意的是 long 类型只存在于 Python2.X 版本中, 在 2.2 以后的版本中, int 类型数据溢出后会自动转为 long 类型。在 Python3.X 版本中 long 类型被移除, 使用 int 替代。

String (字符串) 用于存储文本, 需要注意字符串的下表取值方式为: 从左到右索引默认 0 开始的, 最大范围是字符串长度少 1; 从右到左索引默认 -1 开始的, 最大范围是字符串开头; 同时可以使用 [头下标:尾下标] 来截取相应的字符串, 获取的子字符串包含头下标的字符, 但不包含尾下标的字符。另外, 加号(+) 是字符串连接运算符, 星号(*) 是重复操作。

List (列表) 用 [] 标识, 是 python 最通用的复合数据类型, 其支持字符、

数字、字符串甚至可以包含列表，同时其下标、[头下标:尾下标]、加号运算符及星号运算符与字符串类型一致。

Tuple（元组）用()标识，其与 list 类似，但是 Tuple 不能被二次赋值，相当于只读列表。

字典(dictionary)是除列表以外 python 之中最灵活的内置数据结构类型。列表是有序的对象集合，字典是无序的对象集合。字典用"{ }"标识。字典由索引/键(key)和它对应的值(value)组成。注意：字典仅能通过索引来取值，而没有下标的概念（无序）。例如：dict={'name': 'tom', 'age': 32, 'height': 175}，其中索引/键(key)：'name'、'age'、'height'， 值(value)：'tom'、32、175。

Python 数据类型转换：Python 本身提供了相关的函数，能够将特定的类型转换为其他类型，比如 float(x)可以将变量 x 转换为浮点型。其他函数及使用方式可以查看相关文档。

Python 运算符：

算数运算符：

+, -, *, /, %(取余数), **(幂), //(取整除，向下取整)

比较运算符：

==, !=, <>(不等于), >, <, >=, <=

赋值运算符：

=, +=, -=, *=, /=, %=, **=, //=

逻辑运算符：

and, or, not

成员运算符：

in, not in

身份运算符：

is, is not，用于判断两个对象 id（内存地址）是否相投
优先度：

如下图，在图中越往上运算的优先度越高

运算符	描述
**	指数 (最高优先级)
~ + -	按位翻转, 一元加号和减号 (最后两个的方法名为 +@ 和 -@)
* / % //	乘, 除, 取模和取整除
+ -	加法减法
>> <<	右移, 左移运算符
&	位 'AND'
^	位运算符
<= < > >=	比较运算符
<> == !=	等于运算符
= %= /= //= -= += *= **=	赋值运算符
is is not	身份运算符
in not in	成员运算符
not and or	逻辑运算符